Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1-26. (cancelled)

27. (New) A processor for use with a shared memory system, the processor implementing an instruction set architecture that comprises:

a first class of instructions whose semantics enable access a local memory for the processor without requiring coordination of data updates in a memory address space between the local memory and the shared memory system; and

a second class of instructions whose semantics require coordination of data updates in the memory address space between the local memory and the shared memory system.

28. (New) The processor of claim 27 wherein the first class of instructions includes a first instruction for setting a first value in the local memory at a first address in the memory address space without requiring setting of the first value in the shared memory at the first address.

29. (New) The processor of claim 28 wherein the first instruction comprises a store local instruction that specifies the first value and the first address.

30. (New) The processor of claim 27 wherein the second class of instructions includes a second instruction for setting a first value in the shared memory system at the first address.

31. (New) The processor of claim 30 wherein setting the first value in the share memory at the first address includes storing the first value at a storage location in the shared memory system associated with the first address.

32. (New) The processor of claim 30 wherein the second class of instructions includes a second instruction for setting a value in the shared memory system that was previously set in the local memory by a first instruction in the first class of instructions.

33. (New) The processor of claim 32 wherein the first instruction comprises a store local instruction that specifies the value and the address and the second instruction comprises a commit instruction that specifies the first address.

34. (New) The processor of claim 33 wherein the commit instruction specifies a range of addresses that includes the first address.

35. (New) The processor of claim 30 wherein the second instruction comprises a store instruction that specifies the value and the first address.

36. (New) The processor of claim 27 wherein the first class of instructions includes a third instruction for retrieving a value from the local memory at a first address in the memory address space independently of a value available from the shared memory system at the first address.

37. (New) The processor of claim 36 wherein the third instruction comprises a load local instruction that specifies the first address.

38. (New) The processor of claim 27 wherein the second class of instructions includes a fourth instruction for retrieving a value available from the shared memory system at the first address.

39. (New) The processor of claim 38 wherein first class of instructions includes a third instruction for retrieving a value at a first address in the memory address space such that execution of the fourth instruction prior to execution of the third instruction constrains the third instruction to retrieve a value from the shared memory system at the first address.

40. (New) The processor of claim 39 wherein the third instruction comprises a load local instruction that specifies the first address and the fourth instruction comprises a reconcile instruction that specifies the first address.

41. (New) The processor of claim 41 wherein the reconcile instruction specifies a range of addresses that includes the first address.

42. (New) The processor of claim 38 wherein the fourth instruction comprises a load instruction that specifies the first address.

43. (New) A processor of claim 27 wherein the instruction set architecture further comprises:

a third class of instructions that constrain an execution order of instructions in the first class of instructions and the second class of instructions to constrain an ordering of data access in the shared memory system system.

44. (New) The processor of claim 43 wherein instructions in the third class of instructions includes a fifth instruction that constrains an ordering of an access at a first address in the shared memory system and an access at a second address in the shared memory system.

45. (New) The processor of claim 44 wherein each of the access at the first address and the access at the second address are in a group consisting of an update to a value in the shared memory system and a retrieval of a value from the shared memory system.

46. (New) The processor of claim 45 wherein the fifth instruction comprises a fence instruction that specifies the first address and the second address.

47. (New) The processor of claim 46 wherein the fence instruction specifies the first address and the second address as a same address.

48. (New) The processor of claim 46 wherein the fence instruction specifies at least one of an address range including the first address and an address range including the second address.

49. (New) A processor implementing an instruction set for use in a multiple-processor system with a memory system that includes local memories each accessible by a different processor of the multiple-processor system and a shared memory accessible by multiple of the processors, the instruction set including:

instructions enabling uncoordinated memory access to the local memory system of a processor; and

instructions enabling coordinated memory access by multiple processors.

50. (New) The processor of claim 49 wherein the instructions enabling uncoordinated memory access to the local memory system enable operation using incoherent local memory systems across processors of the multiple-processor system.

51. (New) The processor of claim 49 wherein the local memory system of a processor comprises a cache memory associated with the processor.

52. (New) A method for representing instructions for a processor coupled to a shared memory system:

representing a memory access to the shared memory system using two instructions, one instruction from each of a plurality of classes of instructions including

a first class of instructions that enable access a local memory for the processor without requiring coordination of data updates in a memory address space between the local memory and the shared memory system and

a second class of instructions that require coordination of data updates in the memory address space between the local memory and the shared memory system.

53. (New) The method of claim 52 further comprising ordering execution of instructions that include the two instructions representing the memory access, the ordering enabling execution instructions between the execution of the first and the second of the two instructions.

54. (New) The method of claim 52 wherein the memory access comprises an update access and the two instructions comprise a store local instruction and a commit instruction.

55. (New) The method of claim 52 wherein the memory access comprises a retrieval access and the two instructions comprise a reconcile instruction and a load local instruction.

56. (New) The method of claim 52 wherein the representing of the memory access is performed at a compilation phase.

57. (New) The method of claim 52 wherein the representing of the memory access is performed dynamically by the processor.

58. (New) The method of claim 52 wherein the representing of the memory access is performed dynamically by a memory system coupled to the processor.